REMARKS

Applicant appreciates the time taken by the Examiner to review Applicant's present application. This application has been carefully reviewed in light of the Official Action mailed November 17, 2005. This Reply encompasses a bona fide attempt to overcome the rejections raised by the Examiner and presents amendments as well as reasons why Applicant believes that the claimed invention, as amended, is novel and unobvious over the applied prior art. Accordingly, Applicant respectfully requests reconsideration and favorable action in this case.

Status of the Specification and Claims

Page 11, paragraph [0035], of the Specification is amended herein. Specifically, at line 17, the reference number 330 was replaced by the reference number 320 to correctly refer to an API layer for vendor-specific APIs, as shown in FIG. 3. No new matter is introduced.

Claims 1-26 were pending. Claims 1-26 were rejected. Claims 1, 5-7, 9, 13-15, 17-18, 20, and 23-25 are amended herein. No claim is cancelled herein. Claim 27 is newly added. Support for the amendments to the claims submitted herein can be found in the Specification as originally filed, particularly on page 8, paragraphs [0026]-[0028], page 10, paragraphs [0033]-[0034], page 11, paragraphs [0035]-[0036], and page 12, paragraphs [0037]-[0038]. No new matter is introduced. By this amendment, claims 1-27 are pending.

Drawing Objections

The drawings were objected to for informality. Specifically, the Examiner indicated that Figure 1 should be designated by a legend such as --Prior Art--. A replacement sheet showing Figure 1 designated as Prior Art is submitted herewith. A mark-up sheet showing changes to Figure 1 is concurrently submitted herewith. Accordingly, withdrawal of this objection is respectfully requested.

Rejections under 35 U.S.C. § 101

Claims 17-26 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. The Examiner suggested changing "method" to "computer implemented method" in the preamble to overcome this rejection. Independent claim 17 is amended herein in accordance with the Examiner's suggestion. Dependent claims 18-26 correspondingly incorporate the computer-implemented method of claim 17. Accordingly, withdrawal of this rejection is respectfully requested.

Rejections under 35 U.S.C. § 112

Claims 1-8 and 17-26 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, the Examiner stated that, as to claims 1 and 17, it is uncertain whether "the API" refers to "a public API". Claims 1 and 17 are amended here to particularly differentiate between the public API and the workflow engine API. Accordingly, withdrawal of this rejection is respectfully requested.

Rejections under 35 U.S.C. § 103

Claims 1-2, 9-10, and 17-19 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,516,356 ("Belknap") in view of Applicant Admitted Prior Art (AAPA). The rejections are respectfully traversed for at least the following reasons:

I. Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either explicitly or implicitly in the references themselves or in the knowledge generally available to one of ordinary skill in the art. "The test for an implicit showing is what the combined teachings, knowledge of one of ordinary skill in the art, and the nature of the problem to be solved as a whole would have suggested to those of ordinary skill in the art." *In re Kotzab,* 217 F.3d 1365, 1370, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000). See also *In re Lee,* 277 F.3d 1338, 1342-44, 61 USPQ2d 1430, 1433-34 (Fed. Cir. 2002) (discussing the importance of relying on objective evidence and making specific factual findings with respect to the motivation to combine references); *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

As to claim 17, the combination of Belknap and AAPA fails to establish a *prima facie* case of obviousness at least because no teaching, suggestion, or motivation to do so can be found either explicitly or implicitly in the references themselves or in the knowledge generally available to one of ordinary skill in the art. Belknap appears to be concerned with an application interface to a media server. The cited col. 1, lines 47-49, of Belknap discloses a media manager that incorporates a common API for converting high level generic commands into device level commands for output to a plurality of media devices **25**. Each of the plurality of media devices **25** is associated with a device level interface **22** (col. 3, lines 9-14).

First, "high level generic commands from a computer application" as disclosed by Belknap does not teach or suggest "a set of generic objects of a public API", as set forth in claim 17. In Belknap, the high level generic commands are from a computer application. They are not part of a common API, nor do they represent any functional characteristics common to heterogeneous underlying workflow engines.

Second, "device level interfaces **22**" as disclosed by Belknap does not teach or suggest "workflow engine APIs", as set forth in claim 17. The device level interfaces **22** are hardware interfaces that communicate device level command signals to the media devices **25**. The present application specifically teaches that an application program interface (also known as application programming interface or API) is a set of routines, protocols and/or software tools that can be used to develop software applications (Specification, page 2, paragraph [0005]). A device level interface is completely different from an API, does not have native software objects, and cannot be used to develop software applications.

Third, according to Belknap, the device level commands from processor **10** *to* media devices **25** may correspond, one-to-one, to individual APIs (col. 5, lines 33-37). However, according to Belknap, these individual APIs together *constitute* the common API (col. 3, lines 29-30). In other words, Belknap does not differentiate between individual APIs and the common API. In Belknap, there is only *one* API layer. The plurality of individual APIs *is* the common API. As such, Belknap neither explicitly nor implicitly teaches or suggests "interfacing said public API with said at least two heterogeneous underlying workflow engines through said associated workflow engine API for each of said at least two heterogeneous underlying workflow engines," as recited in independent claim 17.

Moreover, no teaching, suggestion, or motivation to modify Belknap can be found either explicitly or implicitly in AAPA or in the knowledge generally available to one of ordinary skill in the art. As established by AAPA, at the time the invention was made, to implement different workflow engines in the same workflow management system, an organization typically had to develop entirely different sets of process definitions and entirely different sets of applications in order to interface with vendor-specific APIs (page 3, paragraphs [0006]-[0007]). No knowledge generally available to one of ordinary skill in the art to implement otherwise. Thus, even assuming Belknap could be properly combined with AAPA to provide a common API to heterogeneous workflow engines, one of ordinary skill in the art, at the time the invention was made, would have arrived at a common API comprised of a plurality of individual APIs, each corresponding to a workflow engine and each performing a specific function. In other words, to utilize a new workflow engine, the common API according to the combined teachings of Belknap and AAPA would simply include yet another vendor-specific API, which means that different sets of process definitions and applications still have to be developed for different workflow engines. As such, the combination of Belknap and AAPA could not have solved the problems identified in AAPA, further negating any desirability/motivation to modify Belknap with AAPA.

II. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

Independent claim 17, as amended, recites a computer-implemented method for integrating workflow engines comprising:

creating a public application program interface (public API) for at least two heterogeneous underlying workflow engines, wherein the public API comprises a set of generic objects,

wherein said set of generic objects represent functional characteristics common to said at least two heterogeneous underlying workflow engines,

wherein each of the at least two heterogeneous underlying workflow engines is a computer executable application program operable to manipulate content items in accordance with a process definition, and

wherein each of said at least two heterogeneous underlying workflow engines has an associated application program interface (workflow engine API) and a set of native objects;

interfacing said public API with said at least two heterogeneous underlying workflow engines through said associated workflow engine API for each of said at least two heterogeneous underlying workflow engines; and

mapping said set of generic objects to native objects of each of said at least two heterogeneous underlying workflow engines.

As discussed above, Belknap appears to be concerned with providing a common API to a plurality of media devices. As discussed above, the common API according to Belknap does not maintain a set of generic objects and the device level interfaces themselves do not have native objects. Thus, Belknap does not teach or suggest “mapping said set of generic objects to native objects of each of said at least two heterogeneous underlying workflow engines,” as recited in claim 17. Claim 17 is further amended herein to make certain features of the claimed invention more explicit. For example, the public API comprises a set of generic objects that represent functional characteristics common to the heterogeneous underlying workflow engines, each of which has its own workflow engine API. Thus, the combination of Belknap and AAPA does not teach or suggest all the claim limitations as set forth in claim 17. At least for this reason, a *prima facie* case of obviousness has not been established.

Independent claims 1 and 9 are submitted to be patentable over the combination of Belknap and AAPA for similar reasons as submitted above. If an independent claim is nonobvious under 35 U.S.C. § 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Therefore, dependent claims 2-8, 10-16, and 18-26 are also submitted to be patentable over the combination of Belknap and AAPA. Dependent claims 3-8, 11-16, 20-23, and 25-26 were additionally rejected over the combinations of Belknap and AAPA in view of two secondary references. Traversal to the rejections with respect to claims 3-8, 11-16, 20-23, and 25-26 is submitted below.

Claims 5-7, 13-15, 20, and 23-25 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Belknap and AAPA and further in view of U.S. Patent Application Publication No. 20020133635 A1 ("Schechter"). The rejection is respectfully traversed for similar reasons as stated above. In addition, Applicant submits that the combination of Belknap, AAPA, and Schechter neither explicitly nor implicitly teaches or suggests all the claim limitations as set forth in claims 5-7, 13-15, 20, and 23-25.

As to claim 20, like Belknap, Schechter does not appear to be concerned with solving problems related to workflow engines. Also like Blknap, Schechter seems to be directed to interfacing/interacting with devices having different capabilities. The cited page 3, paragraph 29, of Schechter discloses adapters that can transform information from one format to another. Schechter's adapters seem to facilitate interaction with remote mobile devices over a network. Schechter's adapters are not APIs. They are not associated with any particular application or workflow engine, nor do they have generic or native objects of their own. Furthermore, Schechter's adapters transform data. They do not map calls between two API layers (i.e., generic objects of the public API and native objects of workflow engine APIs). Therefore, the lack of proper motivation to combine Belknap, AAPA, and Schechter notwithstanding, the combination of Belknap, AAPA, and Schechter does not teach or suggest all the claim limitations as set forth in claim 20, e.g., "mapping said call from said application to a native call understandable by a native object from said set of native objects."

As to claim 23, Belknap, AAPA, and Schechter, individually and in combination, do not teach or suggest a public API comprising a set of generic objects. Consequently, the combination of Belknap, AAPA, and Schechter does not teach or suggest "wherein said set of generic objects is maintained based upon an industry standard for workflow management." As mentioned on page 4, paragraph [0009], of the Specification, at the time the invention was made, several standards had been developed for the representation and implementation of workflow products interface. However, vendor-specific workflow engines had not adopted these standards. Thus, organizations are left to implement heterogeneous workflow management products. What is missing from AAPA is any teaching related to a public API comprising a set of generic objects. Since neither Belknap nor Schechter teaches or suggests a public API comprising a set of generic objects and such a public API was not in the knowledge generally available to one of ordinary skill in the art at the time the invention was made, the Examiner appears to have made an argument in hindsight that it would have been obvious to one of ordinary skill in the art at the time the invention was made to have recognized that the generic objects of the public API would have to be based upon an industry standard for workflow management.

For similar reasons as submitted above, claims 5-7, 13-15, and 24-25 are submitted to be patentable over the combination of Belknap, AAPA, and Schechter. Accordingly, withdrawal of this rejection is respectfully requested.

Claims 3-4, 8, 11-12, 16, 21-22, and 26 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Belknap, AAPA, and Schechter as applied to claim 20, further in view of U.S. Patent No. 6,647,396 ("Parnell"). The rejection is respectfully traversed for similar reasons as stated above. In addition, Applicant submits that the combination of Belknap, AAPA, Schechter, and Parnell neither explicitly nor implicitly teaches or suggests all the claim limitations as set forth in claims 3-4, 8, 11-12, 16, 21-22, and 26.

As to claims 21-22, the cited col. 3, lines 5-12, of Parnell mentions that workflows can be made to depend on content classification. Parnell appears to be concerned with a content management system **100** that includes an application portion **102**, an API portion **104**, and a database portion **106**. The application portion **102** interacts with the database portion **106** via the API portion **104** (Parnell, col. 3, lines 15-22). Parnell neither explicitly nor implicitly describes a public API comprising a set of generic objects, "wherein said set of *generic* objects further comprises a payload object," as recited in claim 21. Consequently the combination of Belknap, AAPA, Schechter, and Parnell, does not teach or suggest all the claim limitations as set forth in claims 21-22.

For similar reasons as submitted above, claims 3-4, 8, 11-12, 16, and 26 are submitted to be patentable over the combination of Belknap, AAPA, Schechter, and Parnell. Accordingly, withdrawal of this rejection is respectfully requested.

Newly added claim 27

Newly added claim 27 is directed to a computer-implemented method for providing a standardized application program interface between a plurality of software applications and a plurality of workflow engines, said method comprising:

creating and maintaining a public application program interface (public API) comprising a set of generic software objects;

wherein said set of generic software objects represent functional characteristics common to at least two heterogeneous workflow engines;

wherein each of said at least two heterogeneous workflow engines is a computer executable application program operable to manipulate content items in accordance with a process definition;

wherein each of said at least two heterogeneous workflow engines has an application program interface (workflow engine API) associated therewith;

wherein each workflow engine API comprises a set of native software objects; and

wherein workflow engine APIs of said at least two heterogeneous workflow engines are incompatible; and

translating and mapping said set of generic software objects of said public API to and from said set of native software objects of said each workflow engine API, facilitating interoperability of said plurality of software applications.

As established by AAPA, at the time the invention was made, workflow engines typically utilize proprietary, vendor-specific APIs (Spec., page 2, paragraph [0005]). Because each workflow engine can have its own API, interoperability between different applications can be limited (Spec., page 10, paragraph [0034]). New claim 27 presented herein particularly points out that, by creating and maintaining a public API comprising a set of generic software objects and translating and mapping those generic software objects of the public API to and from native software objects of each workflow engine API, embodiments of the invention can facilitate interoperability of a plurality of software applications that programmers can use to write applications consistent with various, heterogeneous underlying workflow engines, achieving technical advantages not reached by the applicable prior art of record. Moreover, newly added claim 27 recites subject matter not reached by applicable prior art of record under 35 U.S.C. §§ 102 and/or 103(a) as discussed above with respect to claims 1-26. In view of the foregoing, new claim 27 is submitted to be patentable and therefore should be allowed.

Conclusion

Applicant has now made an earnest attempt to place the present application in condition for allowance. Other than as explicitly set forth above, this reply does not include any acquiescence to statements, assertions, assumptions, conclusions, or any combination thereof in the Office Action. For the foregoing reasons and for other reasons clearly apparent, favorable consideration and a Notice of Allowance of all pending claims 1-27 is respectfully solicited. The Examiner is invited to telephone the undersigned at the number listed below for discussing an Examiner's Amendment or any suggested actions for accelerating prosecution and moving the present application to allowance.

The Director of the U.S. Patent and Trademark Office is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 50-3183 of Sprinkle IP Law Group.

Respectfully submitted,

Sprinkle IP Law Group
Attorneys for Applicant

Katharina Schuster

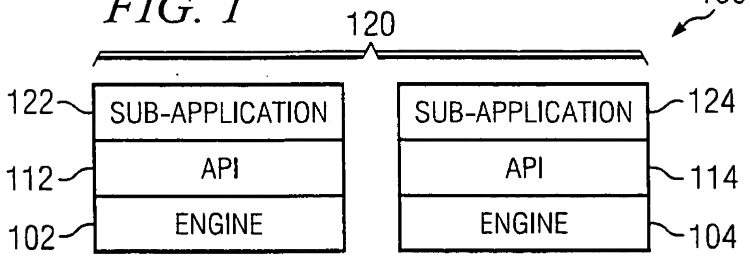
Katharina W. Schuster
Reg. No. 50,000

Date: February 15, 2006

1301 W. 25th Street, Suite 408
Austin, TX 78705
Tel. (512) 637-9229
Fax. (512) 371-9088

FIG. 1
REDLINE CORRECTIONS

FIG. 1



(PRIOR ART)

FIG. 2

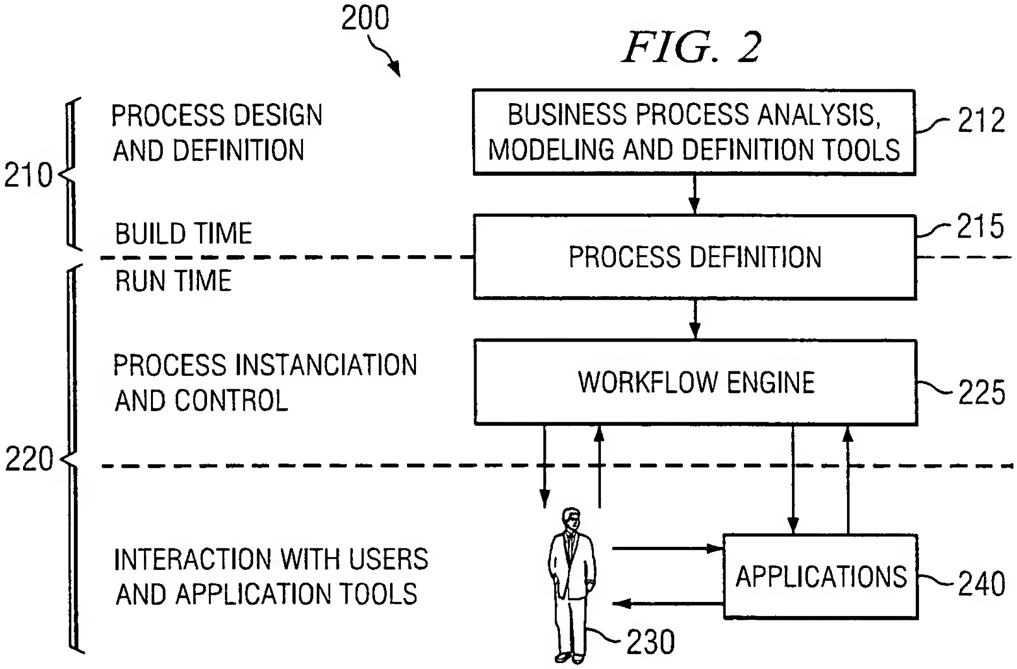
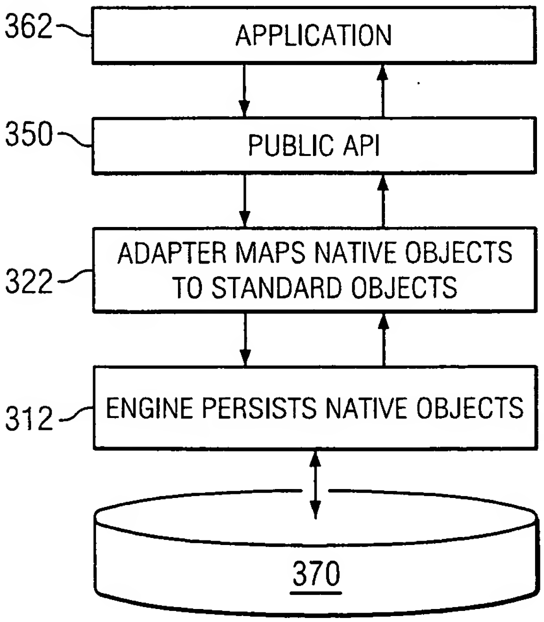

200
210
PROCESS DESIGN AND DEFINITION
BUSINESS PROCESS ANALYSIS, MODELING AND DEFINITION TOOLS
212
BUILD TIME
RUN TIME
PROCESS DEFINITION
215
PROCESS INSTANCIATION AND CONTROL
WORKFLOW ENGINE
225
220
INTERACTION WITH USERS AND APPLICATION TOOLS
APPLICATIONS
240
230

FIG. 4


362
APPLICATION
350
PUBLIC API
322
ADAPTER MAPS NATIVE OBJECTS TO STANDARD OBJECTS
312
ENGINE PERSISTS NATIVE OBJECTS
370